KTH
VETENSKAP
OCH KONST

Royal Institute of Technology

ML 2- MORE HMM & EM ALGORITHM

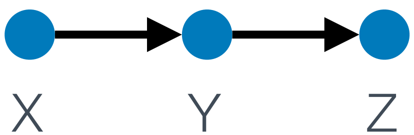
LAST LECTURE

- DGM semantics
- HMMs
  - DP briefly forward, backward etc.
  - Sampling
- Tree DGM marginalization

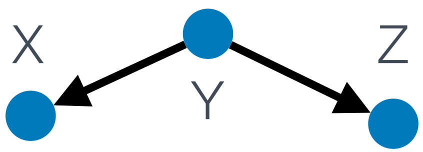
FORKS AND CHAINS IN AN HMM

$$
\begin{tikzcd}
Z_1 \arrow[r] \arrow[d] & Z_2 \arrow[r] \arrow[d] & Z_3 \arrow[r] \arrow[d] & \cdots \arrow[r] & Z_T \arrow[r] \arrow[d] & Z_{T+1} \\
x_1 & x_2 & x_3 & & x_T &
\end{tikzcd}
$$

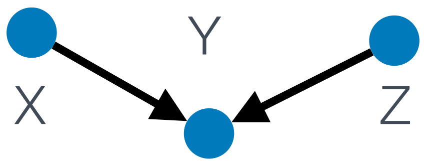
Chain



Fork



v-struct



Applying sum rule

Notice, by the sum rule,

$$f_t(k) = p(x_{1:t-1}, Z_t = k) = \sum_{k' \in [K]} p(x_{1:t-1}, Z_{t-1} = k', Z_t = k)$$

Backward variable

Defined by

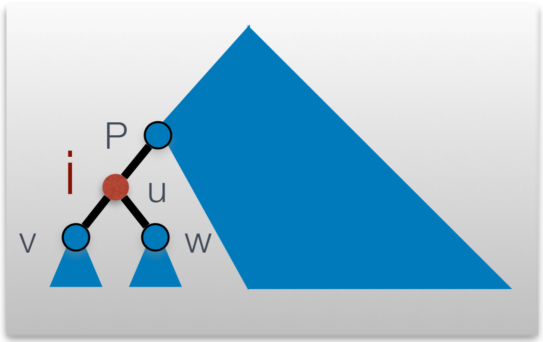
$$b_t(k) := p(\boldsymbol{x}_{t+1:T} | \boldsymbol{Z}_t = k)$$

“Graphical model”

$$\begin{array}{ccccccc} ? & \rightarrow & Z_t = k & \rightarrow & ? & \rightarrow \cdots & ? \\ & & \downarrow & & \downarrow & & \downarrow \\ & & ? & & x_{t+1} & & x_T \end{array}$$

THE MARGINAL

$$P(x_u = i \mid x_O) \propto P(x_u = i, x_O)$$

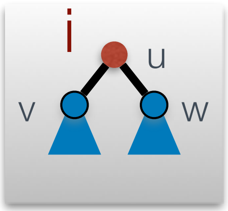


ALGORITHM - MARGINALIZATION TREE DGM

- Given DGM with
  - G=T binary rooted directed tree with vertex set V
  - Bernoulli CPDs
  - Observation x_O, where O is the leaf set
- Compute

$$p(x_O)$$

- Subproblems, subsolutions

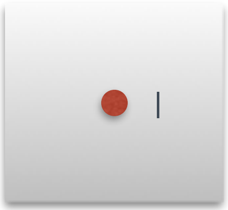


$$s(u,i) = P(x_{\downarrow u \cap O} \mid X_u = i)$$

$x_{\downarrow u}$ variables below u

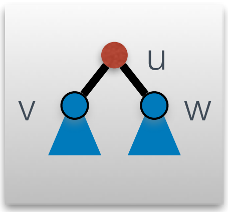
ALGORITHM - MARGINALIZATION TREE DGM

- Visit the vertices of T from leaves to root
  - when at leaf l



$$s(l,i) = \begin{cases} 0 \text{ if } x_l \neq i \\ 1 \text{ if } x_l \neq i \end{cases}$$

  - when at vertex u with children v and w



$$s(u,i) = \left(\sum_{j \in \{0,1\}} P(X_v = j \mid X_u = i) s(v,j) \right) \left(\sum_{j \in \{0,1\}} P(X_w = j \mid X_u = i) s(w,j) \right)$$

CPD for uv   Smaller   CPD for uw   Smaller

THIS LECTURE

- Smoothing
- Sampling
- K-means (inspiration)
- GMM (towards EM)

FILTERING

$$p(\boldsymbol{Z}_t = k|\boldsymbol{x}_{1:t})$$

- Filtering: $p(z_t|x_{1:t})$, online

FILTERING

$$p(\boldsymbol{Z}_t = k|\boldsymbol{x}_{1:t}) = \frac{p(\boldsymbol{x}_{1:t}, \boldsymbol{Z}_t = k)}{p(\boldsymbol{x}_{1:t})}$$

- Filtering: $p(z_t|x_{1:t})$, online

FILTERING

$$p(\boldsymbol{Z}_t = k | \boldsymbol{x}_{1:t}) = \frac{p(\boldsymbol{x}_{1:t}, \boldsymbol{Z}_t = k)}{p(\boldsymbol{x}_{1:t})}$$

$$= \frac{p(\boldsymbol{x}_{1:t-1}, \boldsymbol{Z}_t = k) p(\boldsymbol{x}_t | \boldsymbol{Z}_t = k)}{p(\boldsymbol{x}_{1:t})}$$

- Filtering: $p(z_t|x_{1:t})$, online

FILTERING

$$
\begin{aligned}
p(\boldsymbol{Z}_t = k | \boldsymbol{x}_{1:t}) &= \frac{p(\boldsymbol{x}_{1:t}, \boldsymbol{Z}_t = k)}{p(\boldsymbol{x}_{1:t})} \\
&= \frac{p(\boldsymbol{x}_{1:t-1}, \boldsymbol{Z}_t = k) p(\boldsymbol{x}_t | \boldsymbol{Z}_t = k)}{p(\boldsymbol{x}_{1:t})} \\
&= \frac{f_t(k) p(\boldsymbol{x}_t | \boldsymbol{Z}_t = k)}{p(\boldsymbol{x}_{1:t})}
\end{aligned}
$$

- Filtering: $p(z_t | x_{1:t})$, online

FILTERING

$$
\begin{aligned}
p(\boldsymbol{Z}_t = k|\boldsymbol{x}_{1:t}) &= \frac{p(\boldsymbol{x}_{1:t}, \boldsymbol{Z}_t = k)}{p(\boldsymbol{x}_{1:t})} \\
&= \frac{p(\boldsymbol{x}_{1:t-1}, \boldsymbol{Z}_t = k) p(\boldsymbol{x}_t|\boldsymbol{Z}_t = k)}{p(\boldsymbol{x}_{1:t})} \\
&= \frac{f_t(k) p(\boldsymbol{x}_t|\boldsymbol{Z}_t = k)}{p(\boldsymbol{x}_{1:t})}
\end{aligned}
$$

emission

data probability

- Filtering: $p(z_t|x_{1:t})$, online

OFF-LINE SMOOTHING

$$p(\boldsymbol{Z}_t = k|\boldsymbol{x}_{1:T}) \propto f_t(k)\,\underbrace{p(\boldsymbol{x}_t|\boldsymbol{Z}_t = k)}_{\text{emission}}\,b_t(k)$$

Up to a multiplicative constant

TWO SLICED SMOOTHING MARGINALS - MARGINAL OVER PAIRS OF STATES

$$p(\boldsymbol{Z}_t = k, \boldsymbol{Z}_{t+1} = l | \boldsymbol{x}_{1:T})$$

- Can be computed from forward and backward similarly

TWO SLICED SMOOTHING MARGINALS - MARGINAL OVER PAIRS OF STATES

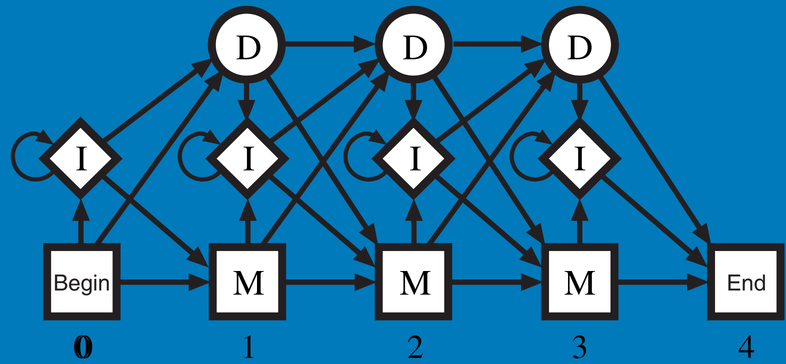
$$p(\boldsymbol{Z}_t = k, \boldsymbol{Z}_{t+1} = l | \boldsymbol{x}_{1:T})$$

$$\begin{array}{ccccccccccccccccccc}
? & \to & ? & \to & ? & \cdots Z_{t-1} = k' & \to & Z_t = k & \to & ? & \to & \cdots & \to & ? & \to & ? \\
\downarrow & & \downarrow & & \downarrow & \downarrow & & \downarrow & & \downarrow & & & & \downarrow & & \\
x_1 & & x_2 & & x_3 & x_{t-1} & & x_t & & x_{t+1} & & & & x_T & &
\end{array}$$

- Can be computed from forward and backward similarly

SAMPLING FROM POSTERIOR

$$z^s_{1:T+1} \sim p(\boldsymbol{Z}_{1:T+1} = k | \boldsymbol{x}_{1:T})$$



$$b_t(k) = \sum_l \underbrace{p(\boldsymbol{Z}_{t+1} = l | \boldsymbol{Z}_t = k)}_{\text{transition}} \underbrace{b_{t+1}(l)}_{\text{"smaller"}} \underbrace{p(\boldsymbol{x}_{t+1} | \boldsymbol{Z}_{t+1} = l)}_{\text{emission}}$$

How much did each previous state contribute to the probability mass of the present state?

BACKWARDS SAMPLING OF POSTERIOR

$$\begin{array}{ccccccccccc} Z_1 & \rightarrow & Z_2 & \rightarrow & Z_3 & \rightarrow & \cdots & \rightarrow & Z_T & \rightarrow & Z_{T+1} \\ \downarrow & & \downarrow & & \downarrow & & & & \downarrow & & \\ x_1 & & x_2 & & x_3 & & & & x_T & & \end{array}$$

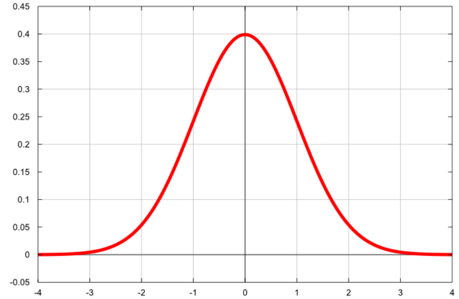
Sample $\quad z_{1:T+1} \sim p(Z_{1:T+1} = k \mid x_{1:T}) \quad$ by

GM DAG $\quad Z_1 \rightarrow Z_2 \rightarrow \cdots \rightarrow Z_{i-1} \rightarrow Z_i \cdots \rightarrow Z_T \rightarrow Z_{T+1}$

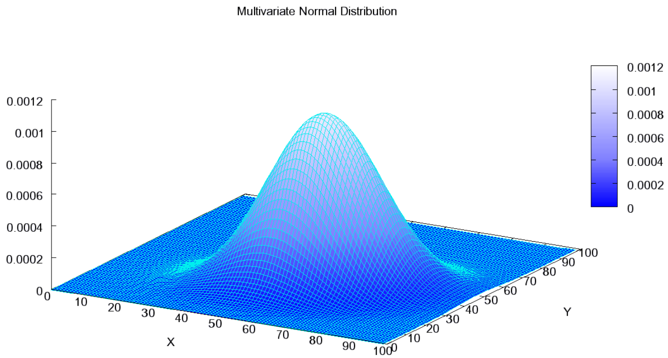
CPDs $\quad p(Z_1 \mid x_{1:T}) \qquad p(Z_i \mid Z_{i-1}, x_{i:T}) \qquad p(Z_{T+1} \mid Z_T)$

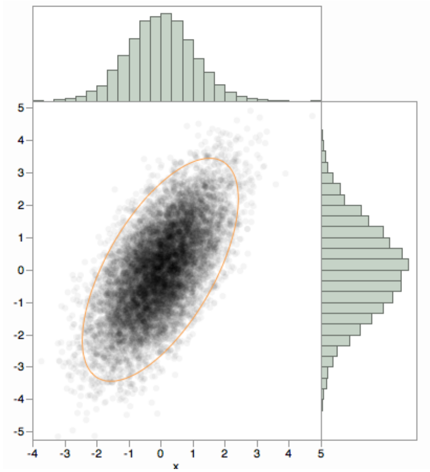
Expectation Maximization (EM)

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu},\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)$$

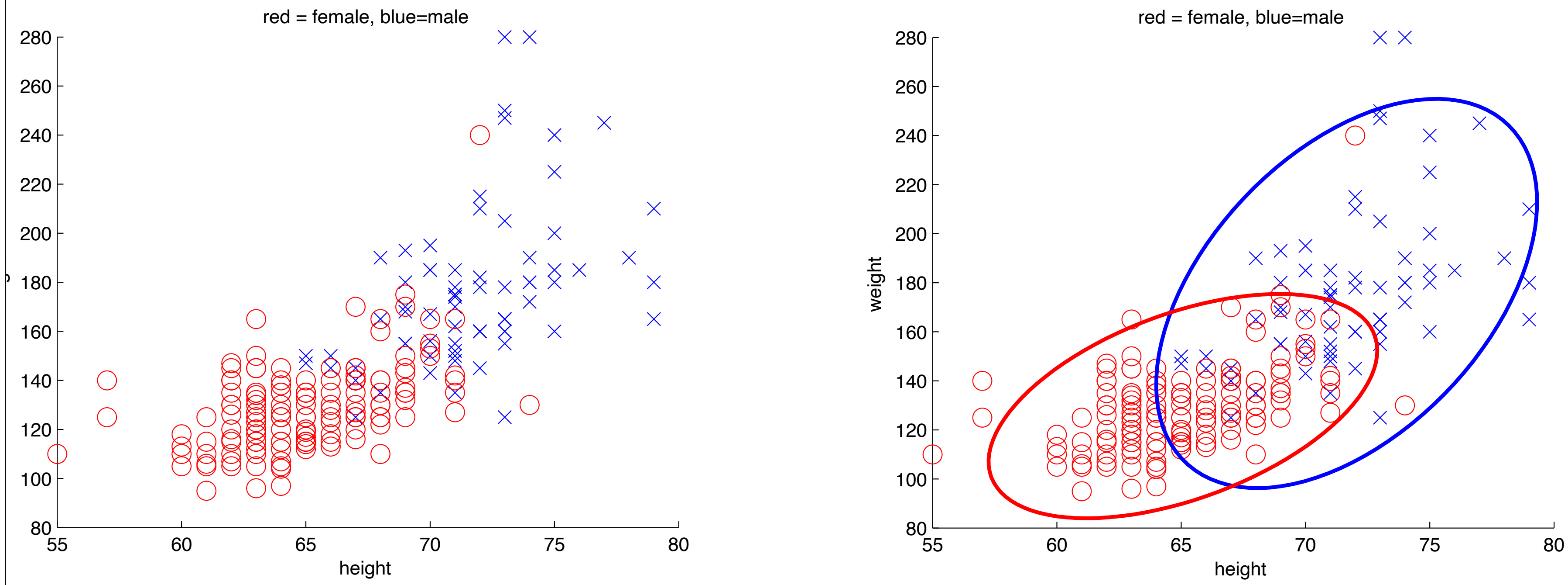


$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$$





GAUSSIAN – MVN



TWO DIMENSIONAL NORMAL

K-MEANS

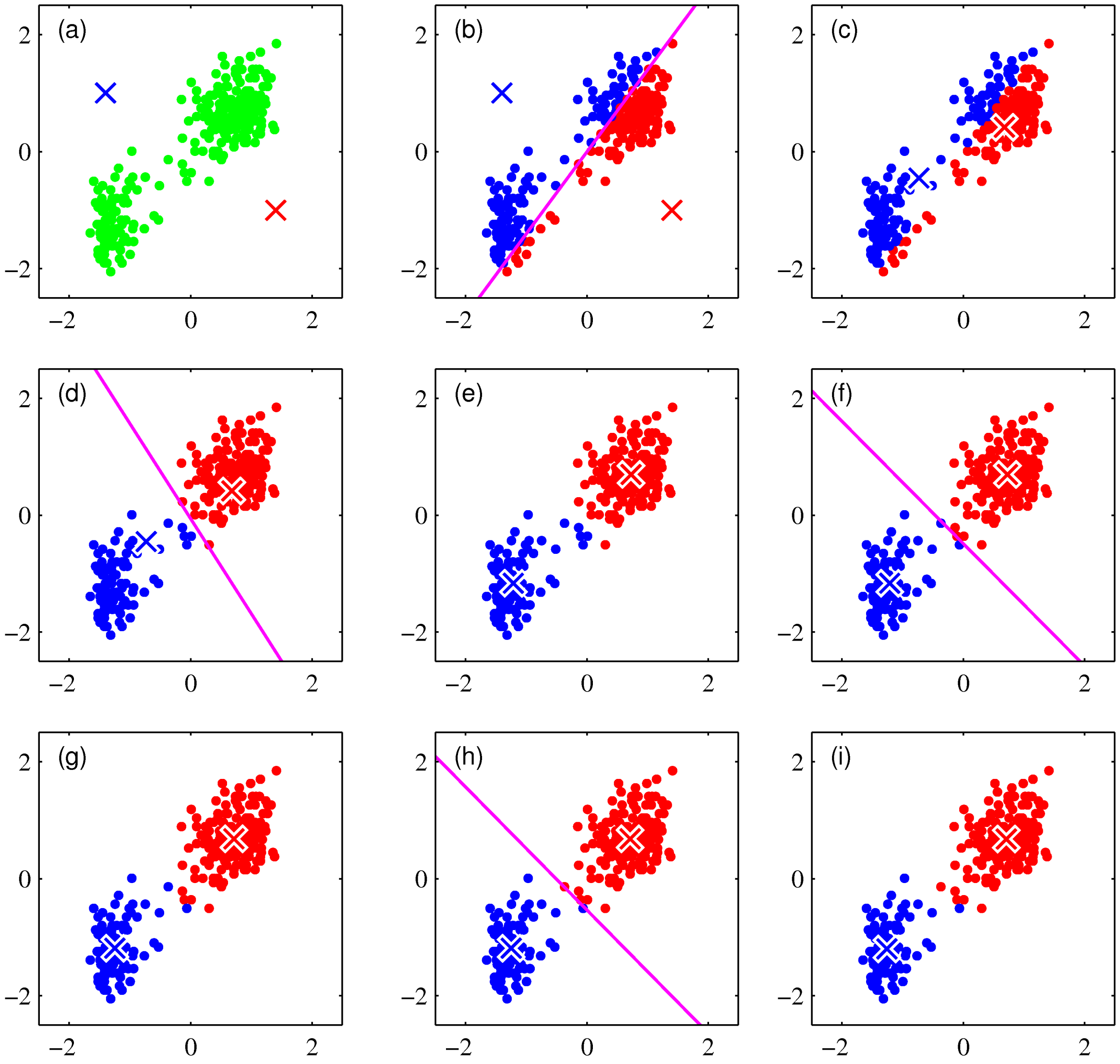
- ★ Data vectors $D=\{x_1,\ldots,x_N\}$
- ★ Randomly selected clusters $z_1,\ldots,z_N$ from C clusters
- ★ Iteratively do

$$\boldsymbol{\mu}_c = \frac{1}{N_c} \sum_{n:z_n=c} \boldsymbol{x}_n, \qquad \text{where } N_c = |\{n : z_n = c\}|$$

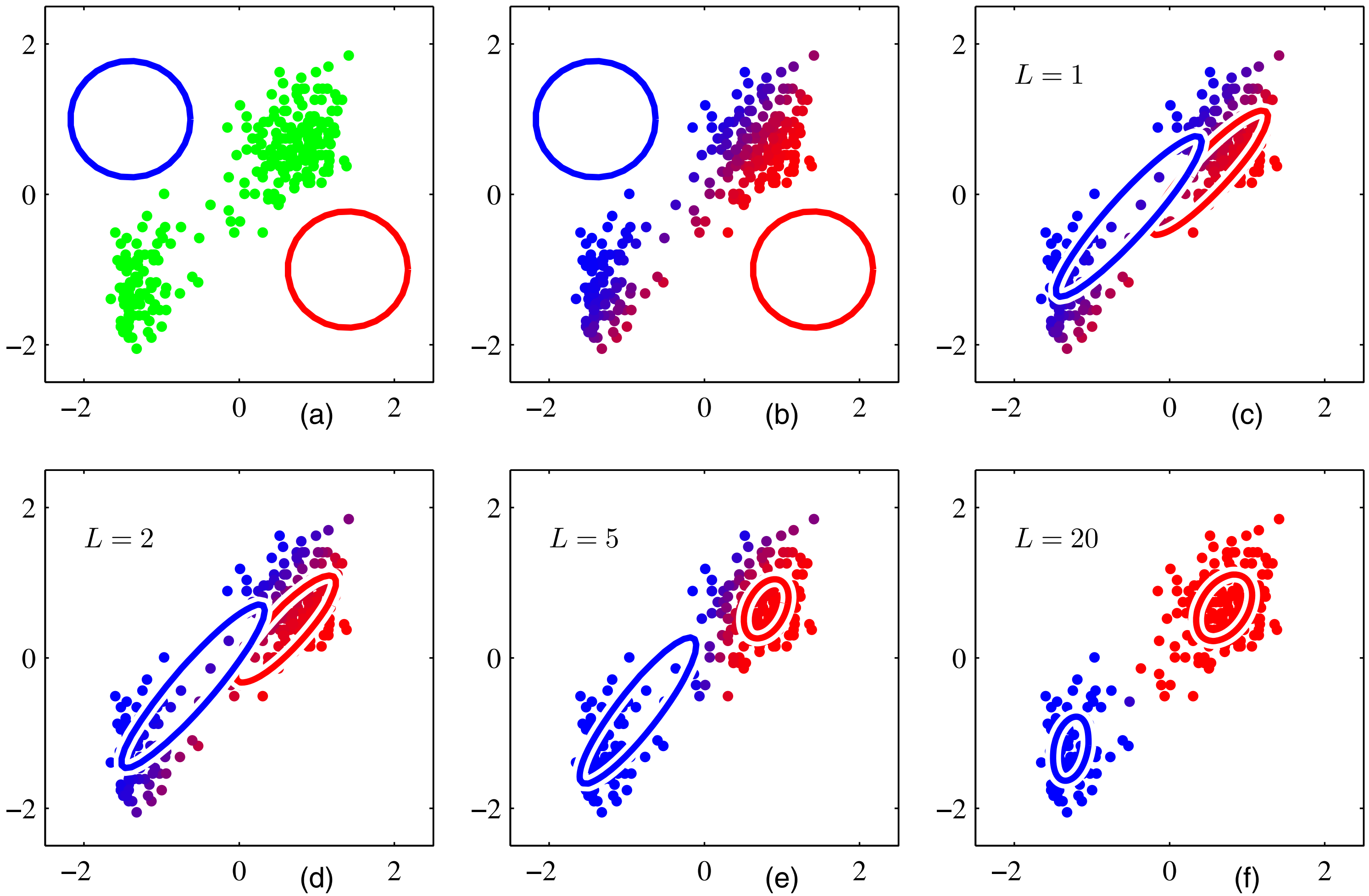
$$z_n = \operatorname{argmin}_c ||\boldsymbol{x}_n - \boldsymbol{\mu}_c||_2$$

- ★ One step O(NKD), can be improved

ASSIGN EACH POINT TO A MEAN



ASSIGNING POINTS TO MULTIPLE MEANS (SOFT)



K-MEANS AS GMM

- Fixed variance, a Gaussian and mean per cluster, i.e., $\theta_c = (\mu_c, \sigma^2)$
- Idea: each point can belong to several means (clusters), generate with categorical
- Use responsibilities to find means

$$r_{nc} = p(z_n = c|\boldsymbol{x}_n, \theta) = \frac{p(z_n = c|\theta)p(\boldsymbol{x}_n|z_n = c, \theta)}{\sum_{c=1}^{C} p(z_n = c|\theta)p(\boldsymbol{x}_n|z_n = c, \theta)}$$

$$\boldsymbol{\mu}_c = \frac{1}{N_c}\sum_n r_{nc}\boldsymbol{x}_n, \qquad \text{where } N_c = \sum_n r_{nc}$$

IMAGE SEGMENTATION WITH K-MEANS

$K = 2$

$K = 3$

$K = 10$

Original image

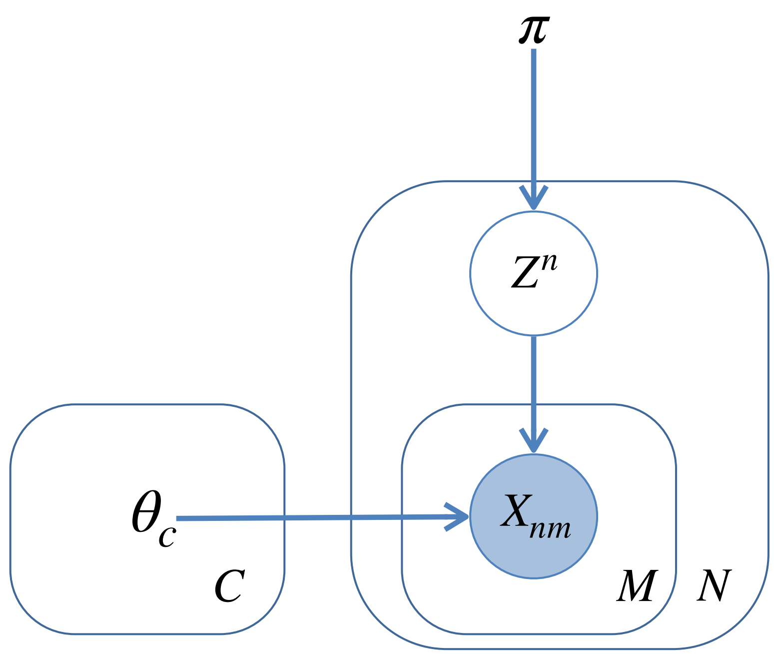
GAUSSIAN MIXTURE MODEL

$\mathcal{D} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ Each a vector

$p(Z = c) = \pi_c$

$p(X \mid Z = c) = \mathcal{N}(X \mid \mu_c, \sigma_c)$

$\boldsymbol{\theta}_c = (\boldsymbol{\mu}_c, \sigma_c)$



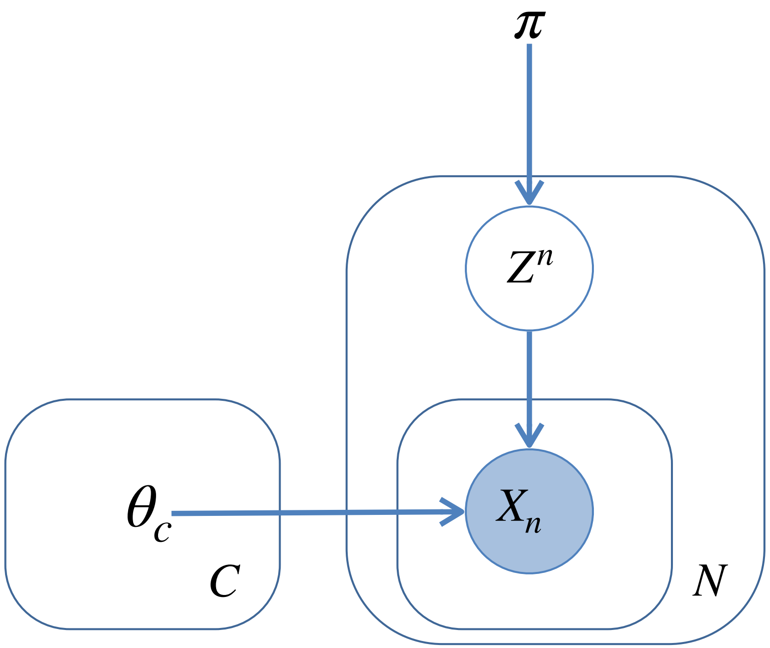
1-DIM GAUSSIAN MIXTURE MODEL

$$\mathcal{D} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$$

$$p(Z = c) = \pi_c$$

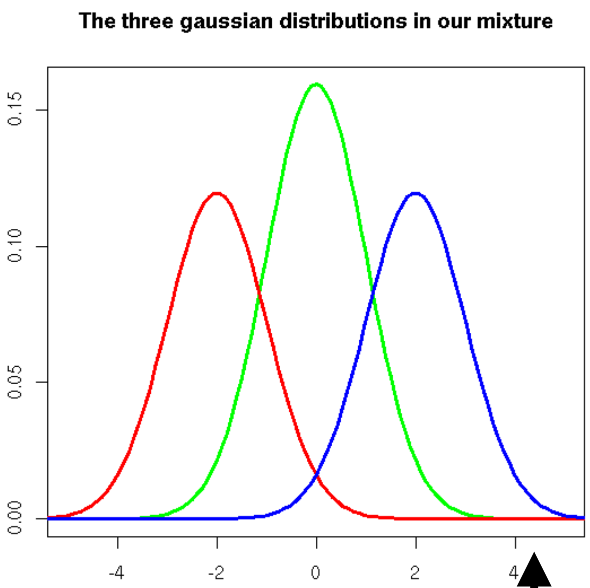
$$p(X \,|\, Z = c) = \mathcal{N}(X \,|\, \mu_c, \sigma_c)$$

$$\boldsymbol{\theta}_c = (\boldsymbol{\mu}_c, \sigma_c)$$



EXAMPLE

z_n is red with probability 1/2, green with probability 3/10, blue with probability 1/5



z_n = blue x_n

z_n is generated as above

x_n is generated from the Gaussian indicated by z_n

We get $x_1,\ldots,x_N$

EM & EXPECTED LOG LIKELIHOOD (Q-TERM)

- Iteratively maximizing the expected log likelihood (expected sufficient statistics).

- Iteratively maximizing the expected log likelihood in practice always leads to a local maxima

- The expectation is over latent variables given data and current parameters

- We maximize the expression by choosing new parameters.

RELATIONS BETWEEN LOG-LIKELIHOODS AND Q-TERMS

log-likelihood

Q-term or expected complete log-likelihood (ECLL)

$$Q(\theta, \theta^i) = \sum_n E_{p(Z_n|x_n,\theta^i)}\left[l(\theta; Z_n, x_n)\right]$$

Theorem: by increasing the ECLL (Q-term), we increase the likelihood.

The ECLL may not increase in every step!

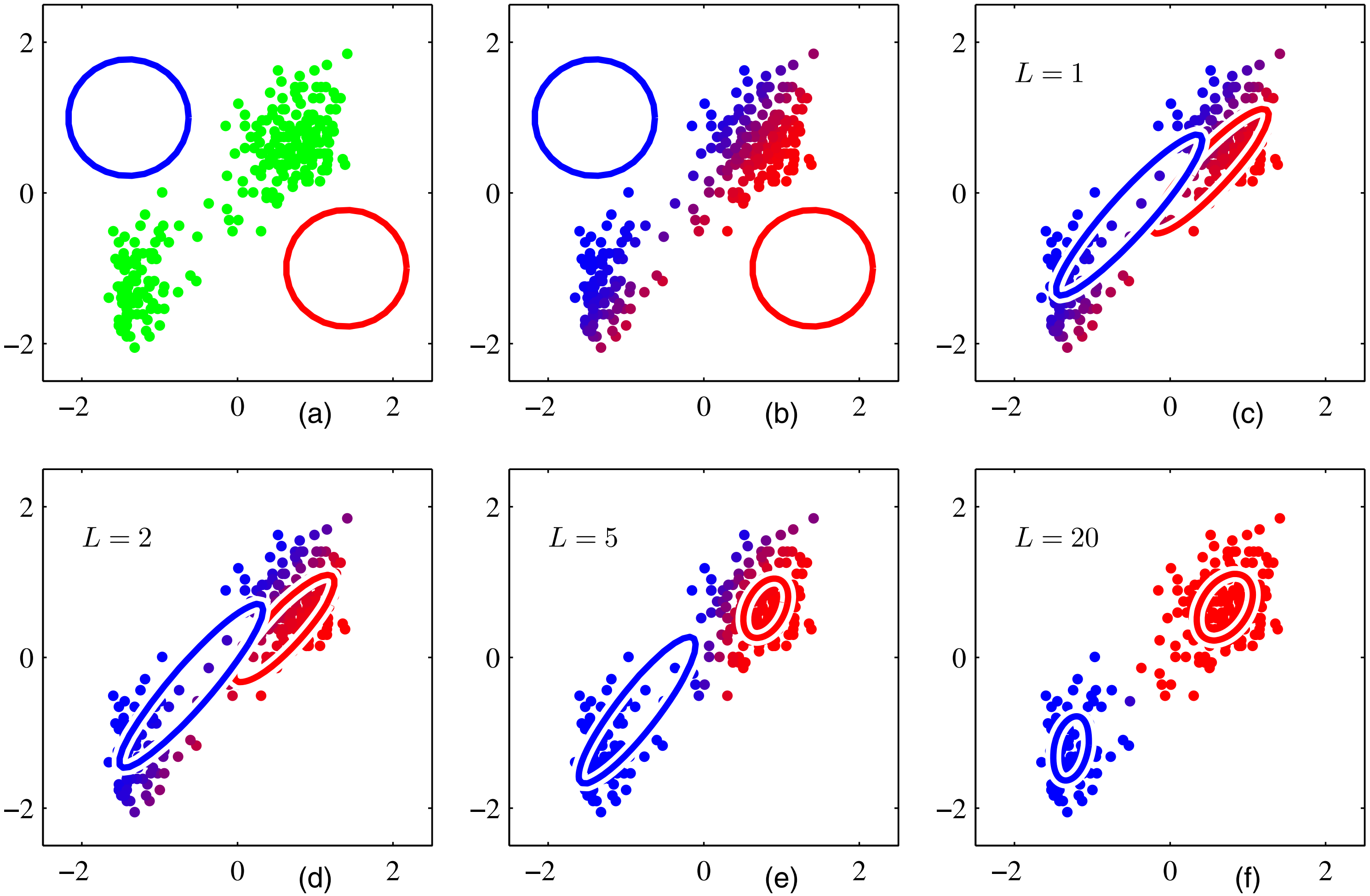
EM-ALGORITHM IN GENERAL

- E-step: compute $E_{p(Z_n|x_n,\theta^i)}\left[l(\theta; Z_n, x_n)\right]$
- M-Step:

$$\theta^i = \text{argmax}_\theta \sum_n E_{p(Z_n|x_n,\theta^i)}\left[l(\theta; Z_n, x_n)\right]$$

- Stop when solution or likelihood hardly change otherwise repeat

E AND M STEPS



GMM EM-ALGORITHM

- E-step: compute $r_{nc} = p(Z_n = c | x_n, \theta^i)$

- M-Step: maximize (1) mixture coefficients and (2) each

$$\sum_n r_{nc} \log \frac{1}{\sqrt{2\pi\sigma_c^2}} \exp\left(-\frac{1}{2\sigma_c^2}(x_n - \mu_c)^2\right)$$

  by setting

$$\mu_c = \frac{\sum_n r_{nc} x_n}{r_c} \quad \text{and} \quad \sigma_c^2 = \frac{1}{\alpha_c^2} = \sum_n r_{nc}(x_n - \mu_c)^2 / r_c$$

- set $\theta^{i+1} = \theta$

- Stop when solution or likelihood hardly change otherwise repeat

- ★ Starting points
- ★ Number of starting points
- ★ Sieving starting points
- ★ The competition
    - The first iterations of EM show huge improvement in the likelihood. These are then followed by many iterations that slowly increase the likelihood. Gradient methods shows the opposite behaviour.

PRACTICAL ISSUES

THE END